

案例 4-直线中点 Bresenham 算法

文档编写：霍波魏

校稿/修订：孔令德

时间 2019~2020

联系方式：QQ997796978

说明：本套案例由孔令德开发，原版本为 Visual C++6.0，配套于孔令德的著作《计算机图形学-基于 MFC 三维图形开发》一书。孔令德计算机工程研究所的学生霍波魏在学习计算机图形学期间，对本套案例进行了升级并编写了学习文档。现在程序的编写和程序的解释都是基于 Windows 10 操作系统，使用 Microsoft visual studio 2017 平台的 MFC（英文版）开发。

一、知识点

1. CLine 类

设计直线类 CLine，成员函数为 MoveTo 和 LineTo 绘制自定义直线。

2. CRGB 类

设计颜色类 CRGB 类，成员变量包括浮点型的 red, green 和 blue。

二、案例描述

本案例通过确定圆心，在一个以坐标原点为圆心的假想圆的圆周上绘制一定数量的点，并在这些点和圆心之间做直线，同时为每条直线圆心处的点和圆周上的点设定一种颜色，使得颜色渐变，最终让其旋转起来。

三、实现步骤

1. 添加 DoubleBuffer()双缓冲函数。
2. 添加 CLine 类，在类中定义绘制直线函数、颜色线性插值函数。
3. 添加 CRGB 类，在类中对运算符进行重载。
4. 在 CTestView.h 中添加一个 DrawObject 函数将假想圆以及各个顶点到圆心的连线。
5. 在 DoubleBuffer()中对 DrawObject()调用。
6. 状态栏的修改以及对 bPlay 进行设置。
7. 在 CTestView.cpp 的构造函数中对参数进行初始化。
8. 在 OnDraw()中对 SetTimer()和 DoubleBuffer()进行调用。

四、主要算法

1. CTestView 类

```
void CTestView::DoubleBuffer(CDC* pDC)//双缓冲
{
    CRect rect;//定义客户区矩形
    GetClientRect(&rect);//获得客户区的大小
    pDC->SetMapMode(MM_ANISOTROPIC);//pDC 自定义坐标系
    pDC->SetWindowExt(rect.Width(), rect.Height());//设置窗口范围
    pDC->SetViewportExt(rect.Width(), -rect.Height());
        //设置视区范围,x 轴水平向右, y 轴垂直向上
    pDC->SetViewportOrg(rect.Width() / 2, rect.Height() / 2);//客户区中心为原点
    CDC memDC;//内存 DC
    CBitmap NewBitmap, *pOldBitmap;//内存中承载的临时位图
    memDC.CreateCompatibleDC(pDC);//创建一个与显示 pDC 兼容的内存 memDC
    NewBitmap.CreateCompatibleBitmap(pDC, rect.Width(), rect.Height());
        //创建兼容位图
    pOldBitmap = memDC.SelectObject(&NewBitmap);//将兼容位图选入 memDC
    memDC.SetMapMode(MM_ANISOTROPIC);//memDC 自定义坐标系
    memDC.SetWindowExt(rect.Width(), rect.Height());
    memDC.SetViewportExt(rect.Width(), -rect.Height());
    memDC.SetViewportOrg(rect.Width() / 2, rect.Height() / 2);
    rect.OffsetRect(-rect.Width() / 2, -rect.Height() / 2);
    Object->DrawObject(&memDC);//向 memDC 绘制图形
    pDC->BitBlt(rect.left, rect.top, rect.Width(), rect.Height(), &memDC,
        -rect.Width() / 2, -rect.Height() / 2, SRCCOPY);
        //将内存 memDC 中的位图拷贝到显示 pDC 中
    memDC.SelectObject(pOldBitmap);//恢复位图
    NewBitmap.DeleteObject();//删除位图
}

void CTestView::DrawObject(CDC * pDC)
{
    for (int i = 0; i < nCount; i++)
    {
        p[i].x = ROUND(nRadius * cos(i * 10 * PI / 180 + angle));
        p[i].y = ROUND(nRadius * sin(i * 10 * PI / 180 + angle));
    }
    for (int i = 0; i < nCount; i++)
    {
        line->MoveTo(pDC, 0, 0, CRGB(1.0, 0.0, 0.0));
        line->LineTo(pDC, p[i].x, p[i].y, CRGB(0.0, 0.0, 1.0));
    }
}
```

五、实现效果

直线中点 Bresenham 算法效果如图 4-1 和图 4-2 所示。

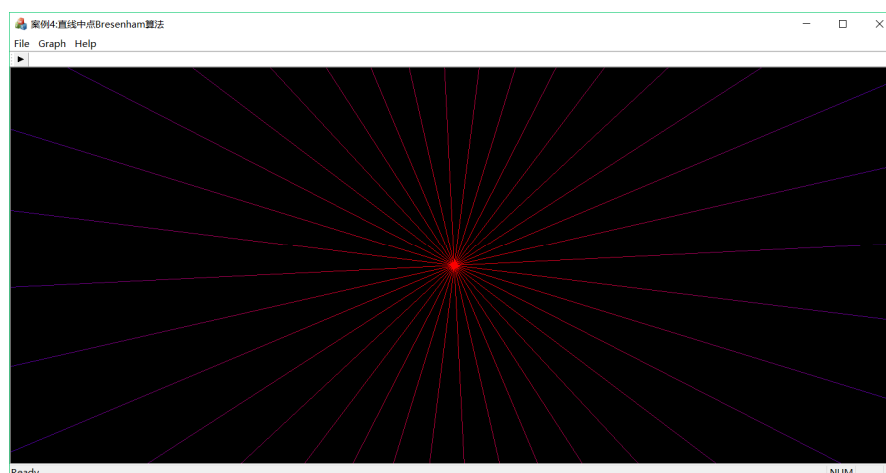


图 4-1 直线中点 Bresenham 算法(双色)效果图

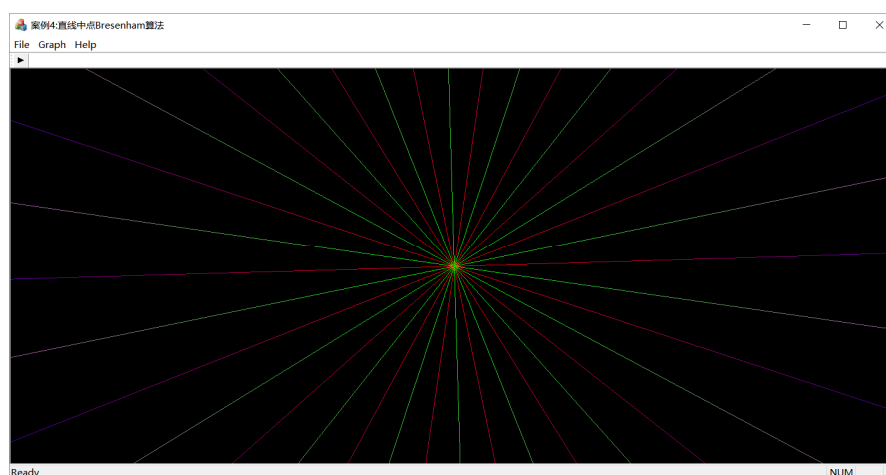


图 4-2 直线中点 Bresenham 算法(四色)效果图

六、遇到的问题及解决方案

1. 添加启动图标按钮

通过设置菜单命令和 Command 命令来控制。

2. 改变旋转角度

在类向导种添加 WM_TIMER 消息映射函数。